Para coletar as métricas de performance, eu programei tanto o backend quanto o script de testes para trabalharem juntos. No backend, cada rota mede o tempo de processamento interno usando performance.now(). Esse valor representa apenas o tempo gasto pelo servidor processando a requisição, sem incluir rede, delay ou tempo de conexão. Esse número é devolvido na resposta como processingTime.

Já o script externo escrito em Node é responsável por medir o tempo total da requisição, iniciando a contagem antes de chamar o endpoint e finalizando após receber a resposta. A diferença entre o tempo total (responseTime) e o processingTime do backend gera a métrica de latência, que indica a parte da demora relacionada à comunicação e não ao servidor em si.

Para testar a escalabilidade, o script executa cada rota com diferentes níveis de requisições simultâneas (1, 5 e 10), criando múltiplas promessas que fazem as chamadas em paralelo. A cada lote, ele calcula a média dos tempos coletados e registra eventuais erros. Todas essas informações são salvas automaticamente em arquivos JSON, um para cada entidade do sistema.

Por fim, esses arquivos de métricas foram usados para gerar gráficos que comparam o comportamento da aplicação em relação ao tempo de resposta, tempo de processamento e latência, permitindo visualizar como o backend reage ao aumento de concorrência.

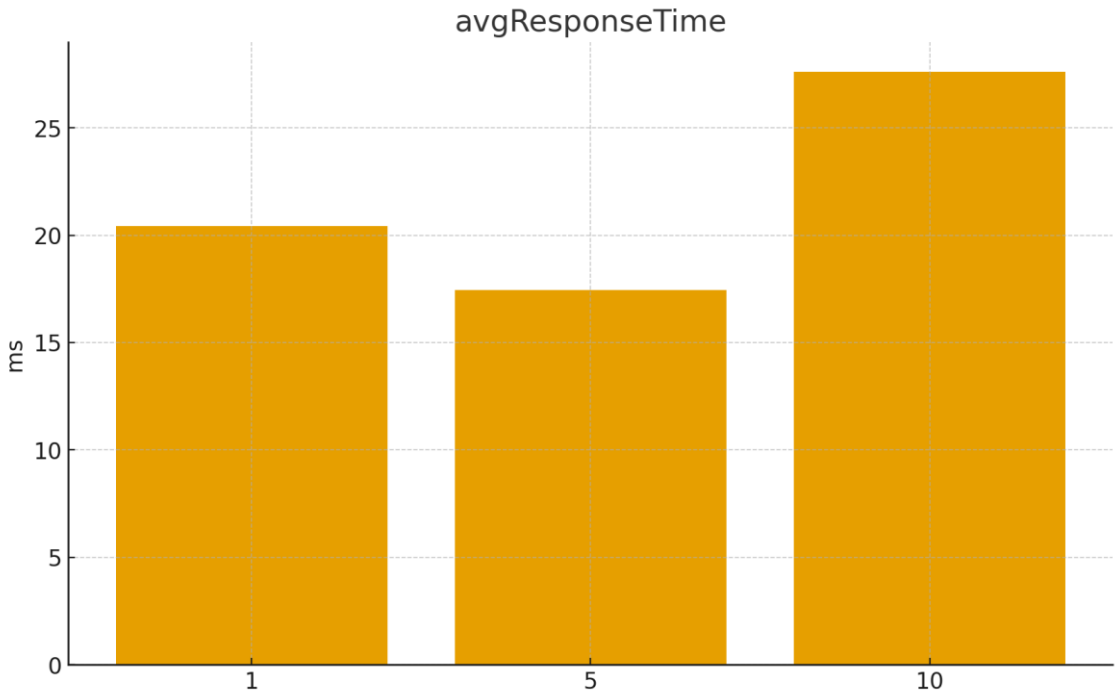

Figura 1 - Gráfico de Tempo Médio de Resposta

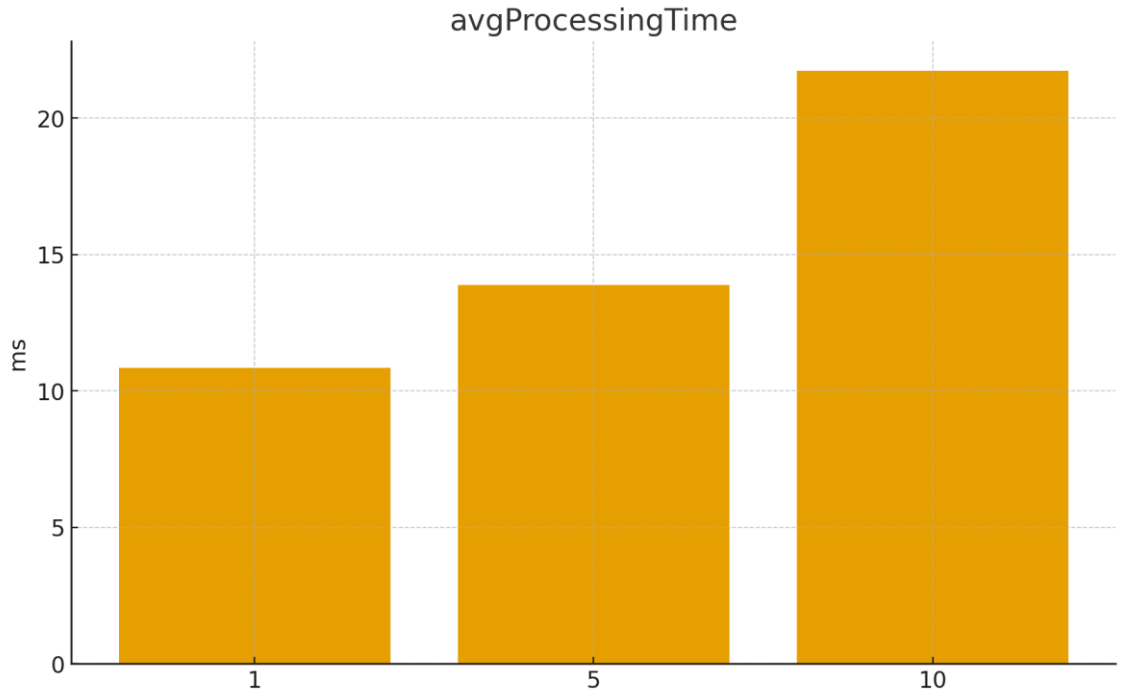

Figura 2 - Gráfico de Tempo Médio de Processamento

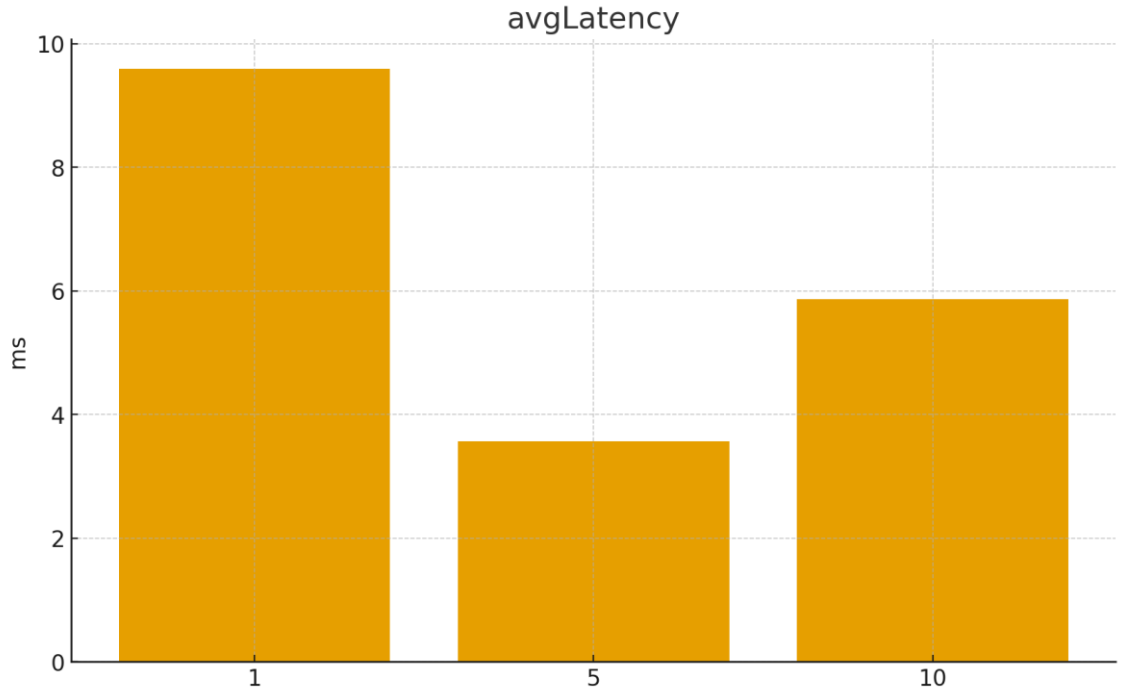

Figura 3 - Gráfico de Média de Latência